Relatório ASA 2º Projeto
Grupo tg042

1. Introdução

No 2º Projecto da cadeira de ASA foi-nos pedido para desenvolver um algorimo que permiti-se a uma empresa de transportes calcular facilmente quais os melhores pontos de encontro nas suas rotas, de forma a que o encontro seja numa localização que retorne o minimo custo possível e que maximize o lucro. O objectivo é então, recebendo informação sobre as rotas,tais como custos e a localização de cada uma das filiais, correr um algoritmo de forma a encontrar o melhor ponto de encontro dessa rota. Isto é feito com algoritmos de caminho mais curto que, tratando as localizações como vertices e as rotas como ligações com peso entre os vertices, calcula o melhor caminho considerando, durante o processo, o grafo respectivo.

Em relação ao programa, o input esperado consiste basicamente no desenho da rede, pelo o que é esperado o nº de localidades, o nº de filiais e o nº de ligações, seguido noutra linha das identificações de cada filial, seguido ainda por a descrição de cada ligação entre localizações. O output esperado é a indicação do melhor ponto de encontro possível, considerando toda a informação e o custo que este encontro terá na empresa. É apresentado também o custo descriminado, de cada uma das filiais que se deslocam até ao ponto de encontro.

Para a codificação do algoritmo utilizámos a linguagem C++, e baseamos o codigo em algoritmos de caminho mais curto, tais como, o Algoritmo de Bellman-Ford, Algoritmo de Dijkstra e Algoritmo de Johnson.

2. Descrição da Solução

Na execução do projecto, realizamos um algoritmo que utiliza como subrotina o Algoritmo de Bellman-Ford.

Para a criação do gráfico, o programa espera pelo input e com este cria uma lista de adjacencias, criando então um grafo virtual com os respectivos pesos.
Em seguida, aplicamos a todos o vertíces filiais, ou seja, os que pertendemos deslocar, o algoritmo de caminho mais curto Bellman-Ford guardando os valores obtidos numa matriz. O algortimo de Bellman-Ford, computa para um dado grafo orientado G com arestas ponderadas, o menor caminho de um nó de origem s até cada um dos outros nós de G. Ao contrário do largamente conhecido algoritmo de Dijkstra, este não impõe nenhuma restrição sobre o sinal do peso das arestas pelo que nos parecia ser a solução mais eficaz.

Após o calculo de todos os dados necessários sobre o respectivo grafo, procedemos a interpretação dos mesmos. Basicamente analiza-mos com um ciclo “for” o custo total de movimentação para cada uma das localidades escolhendo a melhor, ou seja, a de menor peso possível. A partir do momento em que sabemos qual

a localidade ideal, imprimimos os custos de cada uma das filiais como explicado acima no output.

3. Análise Teorica

A complexidade desta resolução é O(N*VE) sendo N o nº de filiais, V o nº de vertices e E o de ligações, sendo a complexidade do algoritmo de Bellman-Ford O(VE). Sabemos que esta solução não é ideal, tanto que esta não passa a todos os testes no mooshak, com erros de memory limit exceeded, pelo que teoricamente, embora responda corretamente aos testes achamos que a complexidade do algoritmo não corresponde á esperada.

Idealmente, a solução passaria por aplicar o algoritmo de Johnson, que consiste em usar como subrotinas tanto Bellman-Ford e Dijkstra reduzindo assim a complexidade da solução.
Isto funciona da seguinte maneira:
1. Adicionar um novo vertice ao grafo e ligá-lo a todos os nós com peso 0;
2. Correr o algoritmo Bellman-Ford nesse novo vertice, considerando as distancias h[0].....h[V-1];
3. Redesenhar o grafo original utilizando como pesos entre u e v : “peso-orginal+h[u]-h[v].
4. Remover o vertice adicionado e correr o algoritmo de Dijkstra ao grafo redesenhado.

Este processo corta a complexidade do algoritmo pois Dijstra é muito mais eficaz nesse sentido, tornando a complexidade deste metodo $O(V^2 \log V + VE)$.

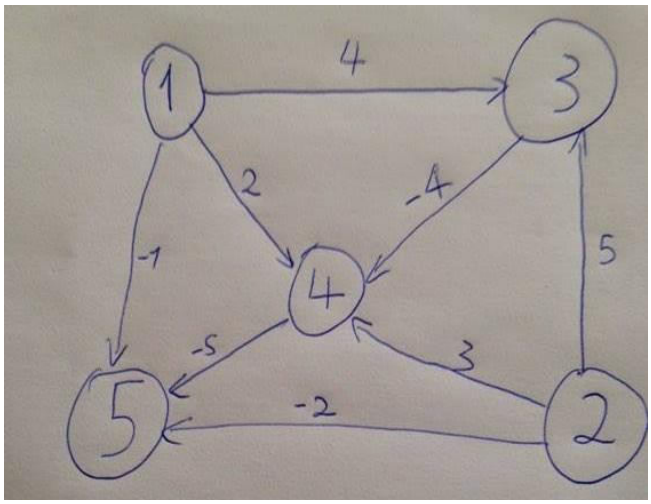
Infelizmente, não conseguimos implementar esta solução no nosso código, pelo que submetemos uma solução menos eficaz.

4. Avaliação Exprimental

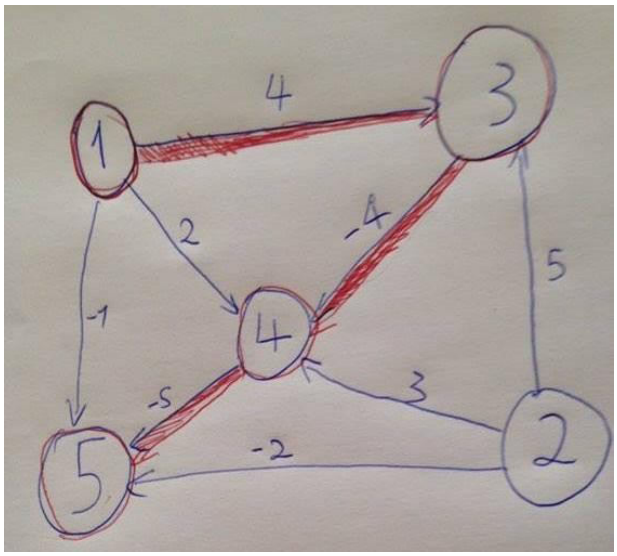
Ao correr os testes disponibilizados verificamos que o algoritmo devolve sempre a resposta correcta, apesar de não passar nos testes 14,15 e 16. Como já foi referido anteriormente, achamos que isto se deve a complexidade da solução apresentada.

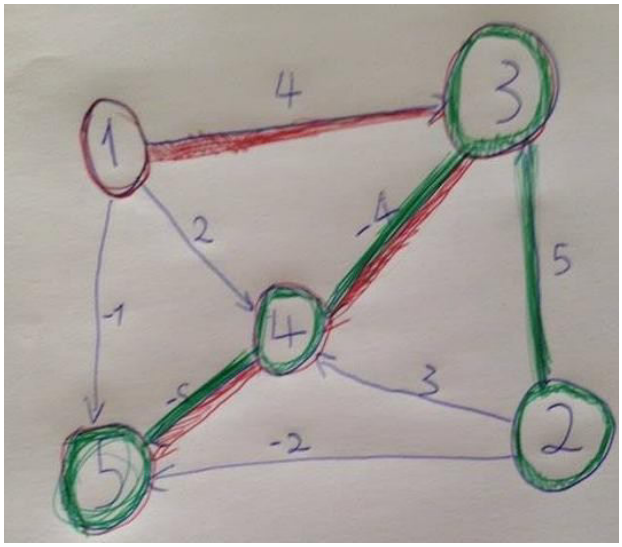
No teste 1, apresentado no enunciado, o processo seria:

```
5 2 8
1 2
4 5 -5
2 4 3
1 4 2
3 4 -4
1 5 -1
2 3 5
1 3 4
2 5 -2
```



1. Desenhar o grafo
2. Aplicar o algoritmo de BellMan-Ford aos filiais 1 e 2. Isto resulta na descrição dos caminhos mais curtos entre o filial e todos os outros vertices.
3.Calculamos o melhor ponto de encontro e apartir daí vemos o percurso necessessário realizar por cada um dos filiais.





Duarte Faria nº79856
Francisco Sousa nº82037